เรื่องน่าอ่าน
ดร.สมนึก คีรีโต

โครงการสมองคอมพิวเตอร์ : นิวรอลเน็ต-เวอร์ก (2)

พลิกโฉมของการใช้คอมพิวเตอร์เพื่อแก้ปัญหาขีดความจำกัดของการใช้งานบางประการ

บทความนี้กล่าวถึงความรู้พื้นฐานเชิงเทคนิคบางประการเกี่ยวกับนิวรอลเน็ตเวอร์ก ซึ่งเป็นโมเดลคอมพิวเตอร์แบบหนึ่งที่ได้รับการคลใจในการออกแบบและสร้างให้เลียนแบบเซลสมองมนุษย์ บทความตอนที่สองนี้จะมุ่งเน้นอธิบายขบวนการในการเรียนรู้ของเน็ตเวอร์ก หลักการของการเรียนรู้ของนิวรอลเน็ตเวอร์กมีหลายวิธีมากมาย ด้วยความจำกัดของหน้ากระดาษในที่นี้จะกล่าวถึงวิธีที่มีชื่อเสียงและกล่าวขวัญถึงกันมากที่สุด นั่นคือ อัลกอริทึมแบบที่มีการส่งผลถอยหลัง (Backpropagation Learning Algorithm)

เกริ่นนำอีกครั้ง

ระบบคอมพิวเตอร์ที่มีระบบการทำงานตามลำดับขั้น (Sequential computers) มีขีดจำกัดในการใช้งานหลายประการ การแก้ไขของปัญหาจำนวนมากกระทำได้ยากในคอมพิวเตอร์แบบลำดับขั้น เช่น ปัญหาที่ต้องเกี่ยวข้องกับการแปลข้อมูลที่มีความซับซ้อน และไม่สามารถกำหนดให้แน่ชัดได้ว่าฟังก์ชันของการแปลเป็นเช่นไร หรือปัญหาที่ต้องมีการคาดเดาคำตอบในขณะที่ข้อมูลอินพุตอาจมีความผิดเพี้ยน หรือเต็มไปด้วยสิ่งรบกวน เป็นต้น

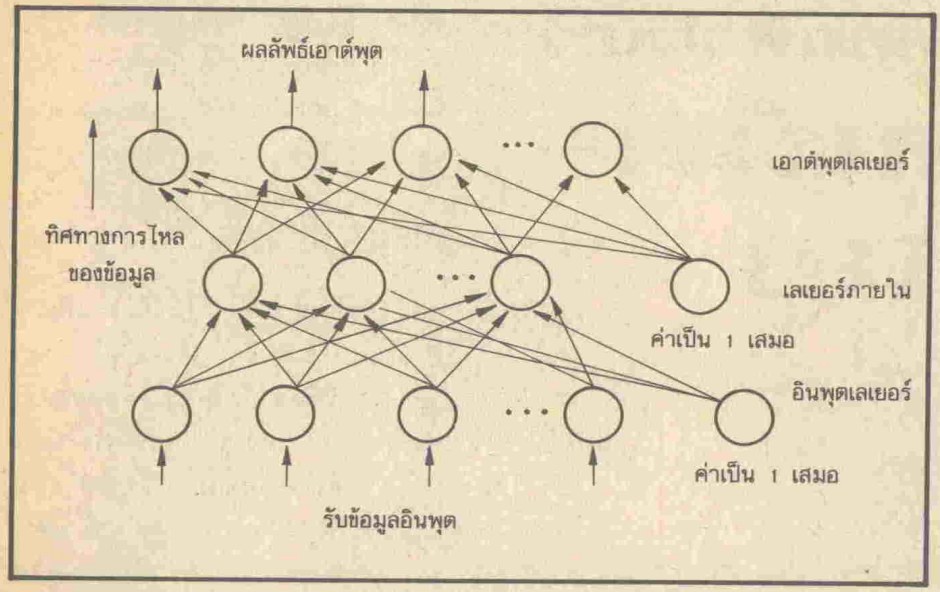
ระบบนิวรอลเน็ตเวอร์กที่จะกล่าวถึงต่อไปนี้ พลิกโฉมของการใช้คอมพิวเตอร์ เพื่อการแก้ไขปัญหาที่กล่าวข้างต้น

เน็ตเวอร์กแบบส่งผลถอยหลัง

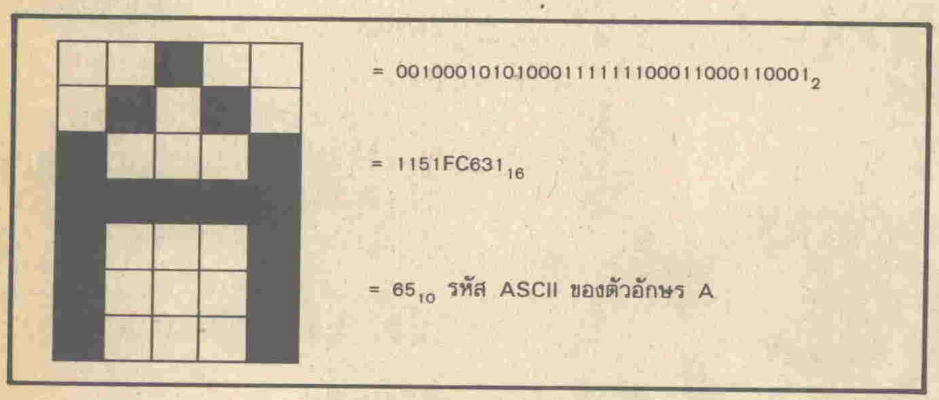
นิวรอลเน็ตเวอร์กแบบที่ใช้งานได้ดีสำหรับการแก้ไขปัญหาที่เกี่ยวกับการรู้จำภาพที่มีความซับซ้อน และปัญหาที่เกี่ยวกับฟังก์ชันการแปลแบบซับซ้อน ก็คือเน็ตเวอร์กแบบส่งผลถอยหลัง หรือ "แบคพรอพาเกชันเน็ตเวอร์ก" (Backpropagation Network)

นิวรอลเน็ตเวอร์กชนิดนี้มีโครงสร้างไดอะแกรม ดังแสดงในรูปที่ 1

ผู้อ่านจะสังเกตได้ว่า เน็ตเวอร์กแบบนี้ประกอบด้วยเซลนิวรอนที่เรียง



รูปที่ 1 โครงสร้างไดอะแกรมของนิวรอลเน็ตเวอร์กแบบที่เรียกว่า "แบคพรอพาเกชันเน็ตเวอร์ก"



รูปที่ 2 ภาพของตัวอักษรหนึ่งตัวในกรอบขนาด 5x7 อาจจะนำมาแปลหรือเทียบให้เป็นรหัสมาตรฐานของแอสกี (ASCII)

กันอยู่อย่างน้อย 3 เลเยอร์ (3 ชั้น) นั่นคือ อินพุตเลเยอร์, เลเยอร์ภายใน และเอาต์พุตเลเยอร์ ทิศทางการไหลของข้อมูลจะเป็นลักษณะเคลื่อนไปข้างหน้าจากอินพุตเลเยอร์ ผ่านเลเยอร์ภายในไปสู่เอาต์พุตเลเยอร์ เน็ตเวอร์กในลักษณะนี้จึงมีชื่อเรียกตามลักษณะโครงสร้างดังกล่าวนี้ว่า "เน็ตเวอร์กหลายชั้นเคลื่อนหน้า" (Multilayer, feedforward network) เน็ตเวอร์กนี้มีความสามารถในการเรียนรู้ในโหมดที่ต้องมีผู้ช่วยภายนอกในการป้อนตัวอย่าง โหมดในการเรียนรู้นี้จึงถูกเรียกในเชิงเทคนิคว่า "การเรียนแบบมีผู้ช่วย" (Supervised Learning)

ตัวอย่างปัญหาต่อไปนี้จะช่วยขยายความของแนวคิดข้างต้น ลองพิจารณาปัญหาของการรู้จำภาพที่ซับซ้อน (Complex Pattern Recognition) โดยเริ่มจากการทดลองออกแบบโปรแกรมคอมพิวเตอร์ที่จะแปลค่าของชุดตัวเลขไบนารีที่มาจากแมตริกซ์ของภาพขนาด 5×7 จุด (ภาพของตัวอักษร) โดยแปลหรือเทียบให้เป็นค่าของรหัสแอสกี (ASCII) ของตัวอักษรตัวนั้น ปัญหาพื้นฐานดังแสดงตัวอย่างในรูปที่ 2 ดูผิวเผินอาจจะเป็นปัญหาง่าย วิธีการที่สะดวกและรวดเร็วที่สุดในการสร้างโปรแกรมก็คือ การใช้หลักการเปิดตาราง (Table Lookup Approach) ตารางที่จะใช้อาจจะเป็นอะเรย์มิติเดียวของเรคอร์ด โดยที่แต่ละเรคอร์ดมีโครงสร้างดังนี้

```
record AELEMENT =
        pattern : long integer ;
        ascii : byte ;
end record ;
```

ฟิลด์แรกของเรคอร์ดเก็บค่าตัวเลขที่แสดงบิตต่าง ๆ ของภาพ (ดูรูปที่ 2 ข้อมูลทั้ง 7 แถวในภาพตัวอักษรถูกนำมาต่อเรียงกัน) ทั้งหมดรวม 35 บิต ฟิลด์ที่สองของเรคอร์ดเก็บตัวเลขรหัสแอสกีของภาพนั้น

อะเรย์มิติเดียวของเรคอร์ดดังกล่าวทำการบรรจุภาพและรหัสแอสกีไว้ทั้งหมด ในส่วนของอัลกอริทึมที่จะทำการรู้จำภาพอินพุตก็เพียงแต่นำแพตเทอร์นของภาพอินพุตมาเปรียบเทียบกับตารางซึ่งเก็บอยู่ในอะเรย์ทีละตัว เรคอร์ดที่มีแพตเทอร์นตรงกันกับอินพุต รหัสแอสกีของเรคอร์ดนั้นก็จะเป็นคำตอบ

แม้ว่าหลักการใช้ตารางนี้จะง่ายและมีความรวดเร็วในการประมวลผล แต่มี

ข้อจำกัดมากในสภาพสถานการณ์จริง ตัวอย่างเช่น ในกรณีที่ภาพอินพุตมีข้อมูลผิดพลาดแม้จะผิดไปเพียงหนึ่งพิกเซล (pixel) อัลกอริทึมดังกล่าวจะให้คำตอบไม่ได้เลย หรือได้คำตอบของรหัสแอสกีที่ผิดได้ ลองนึกถึงกรณีที่มีข้อมูลผิดพลาดที่หลาย ๆ พิกเซล เช่น ข้อมูลผิดเพี้ยนหรือข้อมูลรบกวน (noise) ที่ทำให้ตัวอักษร ตัว "O" มองดูคล้าย "Q" หรือตัวอักษรตัว "E" มองดูคล้าย "F" อัลกอริทึมในการแก้ไขปัญหาด้วยคอมพิวเตอร์แบบลำดับขั้นจะมีความซับซ้อน และต้องใช้เวลาของซีพียูจำนวนมาก สำหรับปัญหาที่กล่าว เรามาลองดูว่าธรรมชาติในการแก้ปัญหาในแบบขนาน (แทนที่จะเป็นแบบลำดับขั้น) ของนิวรอลเน็ตเวอร์กจะช่วยเราเผชิญกับปัญหาเหล่านี้ได้อย่างไร

วัฏจักรการทำงานของแบคพรอพาเกชันเน็ตเวอร์ก

ก่อนที่จะนำนิวรอลเน็ตเวอร์กไปใช้งาน เพื่อการรู้จำ (Pattern Recognition) เราจะต้องทำการฝึกสอนเน็ตเวอร์กก่อนว่าจะรู้จำเรื่องอะไร ขั้นตอนในการเรียนรู้นี้กล่าวคือ เน็ตเวอร์กจะได้รับชุดของตัวอย่างข้อมูลคู่อินพุต-เอาต์พุต (ได้รับการสอนว่า ถ้าอินพุตแบบนี้ เอาต์พุตต้องแบบนี้) สำหรับข้อมูลแต่ละคู่เน็ตเวอร์กจะดำเนินงานของการเรียนรู้เป็นวัฏจักรที่ประกอบด้วย 2 ขั้นตอน คือ "การส่งผลและการปรับค่า" (Propagation and Adaptation) ข้อมูลอินพุตที่ป้อนเข้าสู่นิวรอนในชั้นของอินพุตเลเยอร์จะถูกประมวลผลและส่งผลลัพธ์ต่อไปยังเลเยอร์ชั้นต่อไปจนถึงเอาต์พุตเลเยอร์ ค่าเอาต์พุตที่ได้ (จากการประมวลผลของเน็ตเวอร์ก) จะถูกนำไปเปรียบเทียบกับตัวอย่างเอาต์พุต (ที่กำหนดมา, เอาต์พุตที่ต้องการ) ค่าของความผิดพลาด (ระหว่างเอาต์พุตทั้งสองแบบ) จะถูกคำนวณขึ้นมาสำหรับเซลนิวรอนแต่ละตัว

ค่าของความผิดพลาดจะถูกส่งถอยหลังกลับไปจากเลเยอร์เอาต์พุตไปยังโหนดต่าง ๆ ของเลเยอร์ภายในแต่ละโหนดนั้นจะรับค่าของความผิดพลาดนั้นเพียงบางส่วน ขึ้นอยู่กับว่า โหนดหรือนิวรอนเซลนั้นเป็นตัวที่ส่งผลมากหรือน้อยไปสู่เอาต์พุตนั้น ขบวนการของการส่งค่าความผิดพลาดกลับไปนั้นจะทำซ้ำสำหรับเลเยอร์ถัดลงมาอีก จนกระทั่งทุกโหนดในเน็ตเวอร์กได้รับส่วนแบ่งของค่าความผิดพลาดนั้น หลังจากนั้นน้ำหนักเชื่อมต่อของแต่ละนิวรอนเซลจะถูกปรับค่าไปตามความมากหรือน้อยของสัญญาณค่าความผิดพลาดที่ได้รับ เพื่อที่จะทำให้เน็ตเวอร์กปรับตัวเองไปอยู่ในสถานะที่จะทำให้ตัวอย่างข้อมูลคู่อินพุต-เอาต์พุตได้รับการบันทึกโดยการเข้ารหัสไว้ วัฏจักรของการส่งผลและการปรับค่าจะดำเนินต่อไป จนกระทั่งค่าของความผิดพลาดต่ำกว่าค่าที่กำหนดไว้ค่าหนึ่งแล้วจึงหยุด

จุดสำคัญของขบวนการเรียนรู้ในช่วงจังหวะที่เน็ตเวอร์กได้รับการฝึกฝนนั้นคือ การที่นิวรอลเซลต่าง ๆ ในแต่ละเลเยอร์ทำการปรับค่าน้ำหนักของโหนดแตกต่างกันไป เพื่อที่จะทำการรู้จำลักษณะที่แตกต่างของข้อมูลอินพุต ดังนั้นหลังจากที่ได้รับการฝึกฝนแล้ว เมื่อจะนำเน็ตเวอร์กไปใช้งานเพื่อรู้จำข้อมูลอินพุตชุดหนึ่ง ซึ่งแม้ว่าจะมีสัญญาณรบกวนหรือมีข้อมูลที่ไม่สมบูรณ์ หน่วยต่าง ๆ ของเลเยอร์ภายในจะให้ผลลัพธ์เอาต์พุตได้ถูกต้อง ถ้าอินพุตที่ป้อนเข้ามามีลักษณะใกล้เคียงกับแพตเทอร์นที่เคยเรียนรู้ไปแล้ว ในทางตรงข้ามถ้าลักษณะข้อมูลไม่มีความใกล้เคียงกับแพตเทอร์นที่เคยผ่านการฝึกฝน หน่วยนิวรอนหน่วยนั้นก็จะไม่ส่งผลลัพธ์เลย

สรุปขั้นตอนของการฝึกสอนสำหรับนิวรอลเน็ตเวอร์กแบบส่งผลถอยหลัง (Backpropagation Network) นี้ โดยเขียนเป็นฟอร์มของโปรแกรมดังต่อไปนี้

begin

while ค่าความผิดพลาดของเน็ตเวอร์กมีค่ามากกว่าที่กำหนดไว้

begin

1. รับข้อมูลตัวอย่างอินพุต
2. ทำการประมวลผล (propagate forward) โดยส่งผลลัพธ์ไปยังเอาต์พุตเลเยอร์
3. ทำการคำนวณค่าความผิดพลาดระหว่างค่าเอาต์พุตที่ได้จากเน็ตเวอร์กกับข้อมูลตัวอย่างเอาต์พุตที่ต้องการ
4. ส่งค่าความผิดพลาดกลับเข้าไปในเน็ตเวอร์ก (error backpropagate)
5. ทำการปรับค่าน้ำหนักมากหรือน้อยตามค่าของความผิดพลาด

end

end.

หลังจากเน็ตเวอร์กได้รับการฝึกฝนแล้วก็พร้อมที่จะนำไปใช้งาน ขั้นตอนของการประยุกต์ใช้งานจะสามารถเขียนสรุปในฟอร์มของโปรแกรมดังนี้

begin

1. ทำการรับข้อมูลอินพุตของปัญหา

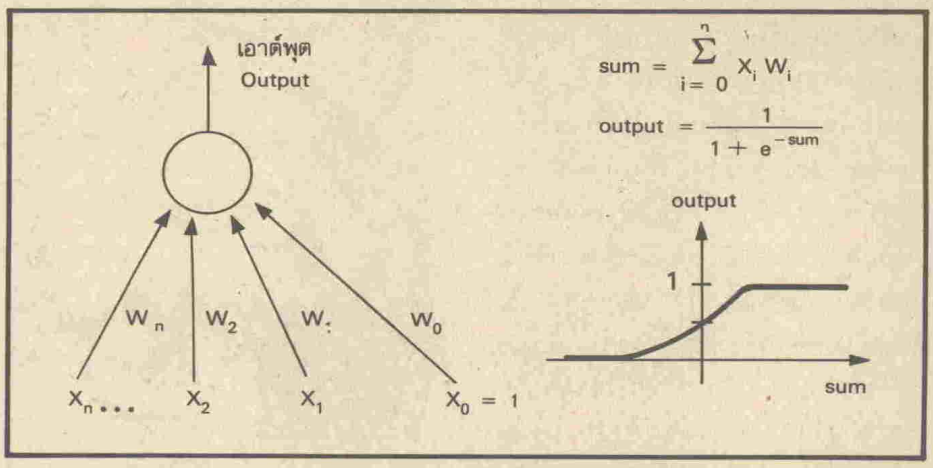
2. ทำการประมวลผล (propagate forward) โดยส่งผลลัพธ์ไปยังเอาต์พุตเลเยอร์

3. ให้คำตอบที่เอาต์พุตเลเยอร์

end.

ในบทความตอนที่ 1 ได้บรรยายขบวนการพื้นฐานของการประมวลผล (Propagate) ของเซลนิวรอนแต่ละเซลไปแล้ว ในที่นี้จะขอสรุปการทำงานของแต่ละโหนดวงกลม (นิวรอนหนึ่งเซล) ของรูปที่ 1 ว่าค่าเอาต์พุตของแต่ละโหนดเป็นฟังก์ชันของค่าอินพุตดังแสดงในรูปที่ 3

สำหรับผู้อ่านที่ติดตามอ่านบทความในตอนที่ 1 ในฉบับที่ 88 มาแล้ว จะพบว่าเอาต์พุตฟังก์ชันของแต่ละเซลตามที่แสดงเป็นสมการในรูปที่ 3 นี้ซึ่งเป็นเหมือน S ฟังก์ชัน (เรียกว่า เป็น Sigmoid ฟังก์ชัน) แตกต่างจากสมการเอาต์พุตของนิวรอลเน็ตเวอร์กแบบเพอเซพตรอน ซึ่งเป็นไบนารีฟังก์ชัน (เอาต์พุตเป็น 0 หรือ 1 เท่านั้น) เหตุผลสำคัญของการใช้ฟังก์ชันแบบ Sigmoid ดังกล่าวในแบคพรอทเกชันเน็ตเวอร์กก็เพื่อที่จะทำให้การคำนวณหาค่าความผิดพลาดและการส่งค่าความผิดพลาดกลับเข้าไปในเน็ตเวอร์ก (Error Back-progate) เป็นไปได้ง่ายขึ้น ซึ่งเป็นเหตุผลในเชิงคณิตศาสตร์ที่ไม่สะดวกต่อการกล่าวถึงรายละเอียดในบทความนี้



รูปที่ 3 โมเดลการประมวลผล (Propagate) ของนิวรอนแต่ละเซลของแบคพรอพาเกชันเน็ตเวอร์กในรูปที่ 1

การประยุกต์ใช้งานนิวรอลเน็ตเวอร์ก

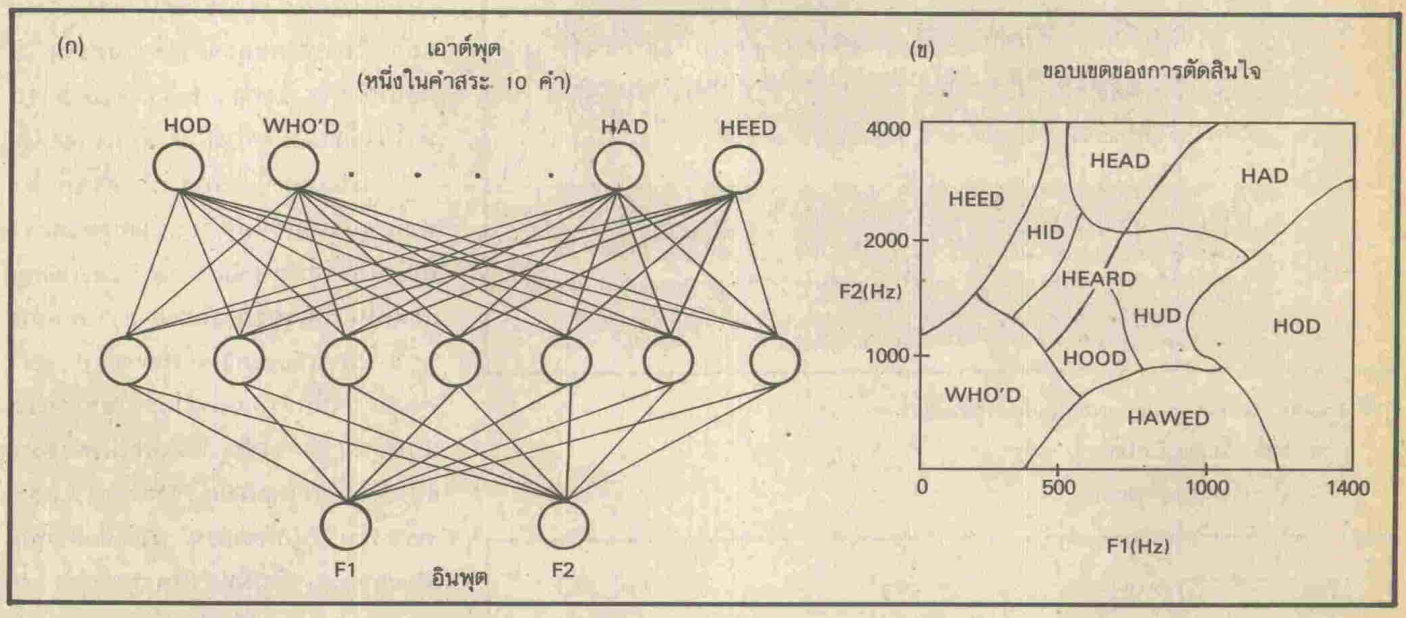
การประยุกต์ใช้งานระบบเครือข่ายนิวรอนแบบแบคพรอพาเกชันที่ประสบความสำเร็จจำนวนมาก มีลักษณะเป็นแบบการใช้งานเพื่อการระลึก หรือการรับรู้ (Pattern Recognizers) ทั้งด้านภาพและด้านเสียงในที่นี้จะขอยกตัวอย่างของทั้งสองกลุ่มดังนี้

การฟังและการพูดด้วยนิวรอลเน็ตเวอร์ก

การที่จะทำให้คอมพิวเตอร์มีความสามารถในการฟังและเข้าใจภาษาพูดเป็นเรื่องที่ยุ่งยากซับซ้อน อย่างไรก็ตามระบบนิวรอลเน็ตเวอร์กถูกนำไปประยุกต์ใช้และแก้ไขปัญหาหลายประการสำหรับการฟังและเข้าใจภาษาพูด (Speech Recognition) รูปที่ 4 แสดงนิวรอลเน็ตเวอร์กแบบสามเลเยอร์นี้ได้รับการฝึกสอนให้มีความสามารถในการแยกแยะระหว่างเสียงของสระที่แตกต่างกันของคำสระ 10 คำ ดังนั้น หลังจากที่เน็ต-เวอร์กนี้ได้รับอินพุตซึ่งเป็นความถี่ของเสียง 2 ความถี่ ผลลัพธ์เอาต์พุตจะเป็นสัญญาณที่บอกว่าเสียงนั้นเป็นคำสระคำไหน (หนึ่งใน 10 คำสระ)

ปัญหาในการแปลประโยคข้อความให้เป็นคำพูด (Speech Production) เป็นอีกปัญหาหนึ่งที่น่าสนใจ ระบบนิวรอลเน็ตเวอร์กก็ถูกนำไปใช้งานด้านนี้อย่างมีประสิทธิภาพ การพูดภาษาด้วยคอมพิวเตอร์เป็นปัญหาที่ง่ายกว่าการที่จะทำให้คอมพิวเตอร์เข้าใจภาษาพูดมาก ตัวอย่างของระบบเครือข่ายนิวรอนที่ชื่อว่า "NETtalk" เป็นระบบที่มีความสามารถในการเรียนรู้เพื่อจะพูดตามข้อความในประโยคภาษาอังกฤษ

นักภาษาศาสตร์ให้ความสนใจและศึกษาถึงกฎเกณฑ์ต่าง ๆ สำหรับการแปลประโยคข้อความให้เป็นภาษาพูด (การศึกษาทางด้านนี้ เรียกว่า "phonemes") ตัวอย่างเช่น ตัวอักษรตัว "x" โดยปกติจะอ่านออกเสียงด้วย "ks" เช่นในคำว่า "box" และ



รูปที่ 4 ตัวอย่างนิวรอลเน็ตเวอร์กแบบสามเลเยอร์ที่สามารถแยกแยะเสียงสระโดยรับอินพุตเป็นความถี่เสียงสองความถี่

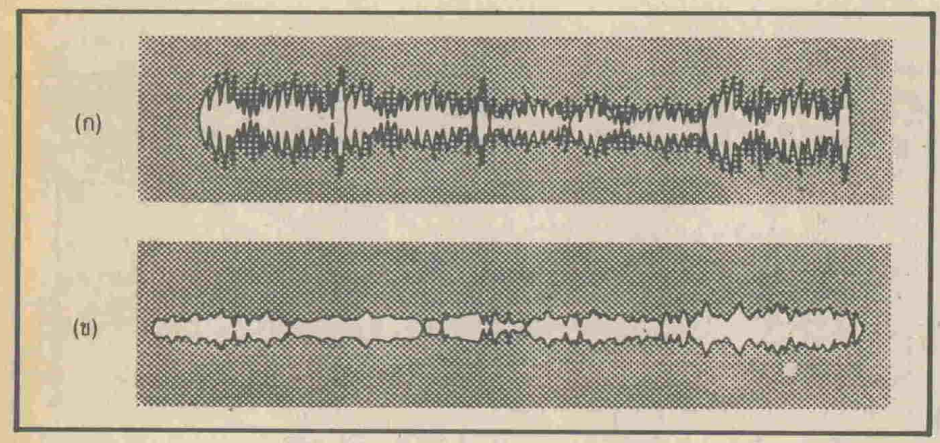
"axe" เป็นต้น อย่างไรก็ตาม ภาษาศาสตร์มักจะมีข้อยกเว้นสำหรับกฎเกณฑ์จำนวนมาก เช่น คำว่า "xylophone" ("x" มิได้อ่านออกเสียง "ks") เป็นต้น ดังนั้น ถ้าเราจะออกแบบและสร้างโปรแกรมคอมพิวเตอร์สำหรับงานด้านนี้ โดยวิธีการปกติเราจะต้องป้อนกฎเกณฑ์ต่าง ๆ ทางภาษาศาสตร์นี้พร้อมกับข้อยกเว้นดังตัวอย่างที่กล่าวจำนวนมาก ปัญหาที่สำคัญคือ กฎเกณฑ์และข้อยกเว้นเหล่านี้อาจจะมีความสัมพันธ์หรือความขัดแย้งกันในหลายจุด ซึ่งทำให้การพัฒนาระบบเป็นไปด้วยความยุ่งยาก ระบบนิวรอนเสนอแนวทางในการแก้ปัญหานี้ในอีกรูปแบบหนึ่งกล่าวคือ นิวรอลเน็ตเวอร์กจะเรียนรู้กฎเกณฑ์และข้อยกเว้นต่าง ๆ เหล่านี้เองด้วยการฝึกฝนด้วยตัวอย่าง "คำ" และ "การออกเสียง" ของคำนั้นจำนวนมากพอ ระบบ NETtalk เป็นนิวรอลเน็ตเวอร์กแบบแบคพรอพาเกชัน ซึ่งสามารถทำงานได้ดีในลักษณะปัญหาดังกล่าวนี้

นิวรอลเน็ตเวอร์กกับการมองเห็น

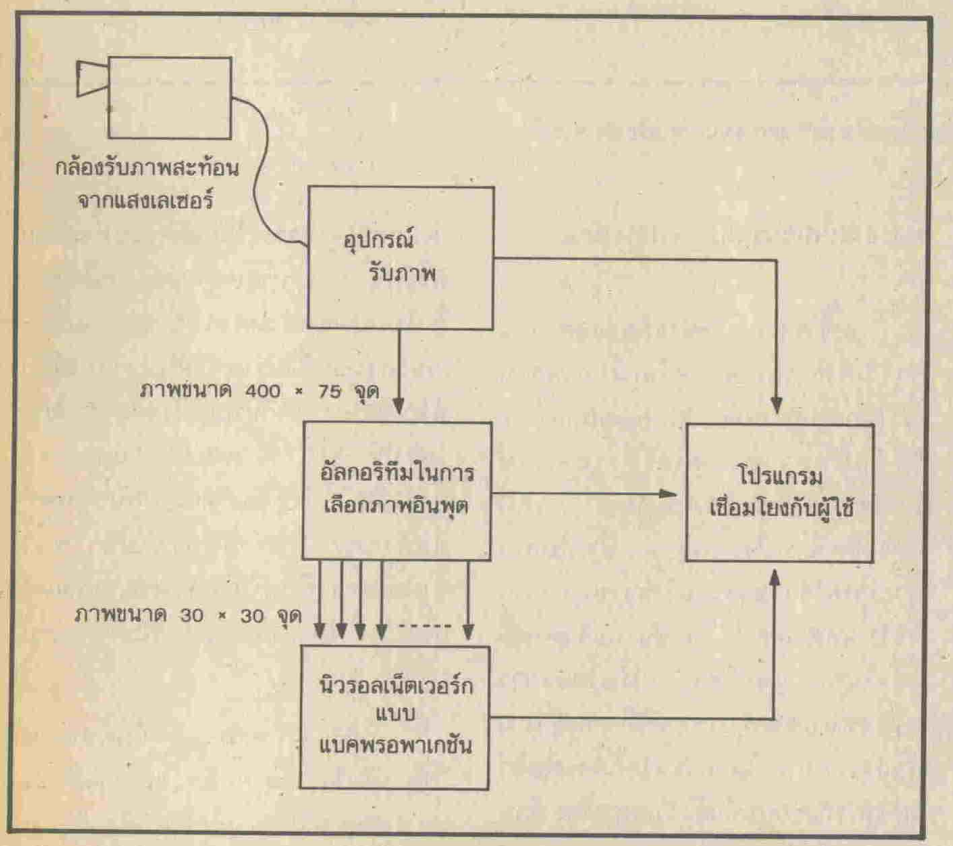
การใช้งานเครือข่ายนิวรอลสำหรับการวิเคราะห์ภาพ หรือการทำความเข้าใจภาพ (Pattern Recognition) ถือได้ว่าเป็นการประยุกต์ใช้งานหลักที่ประสบความสำเร็จมากที่สุด ในที่นี้จะขอยกอีกหนึ่งตัวอย่าง ซึ่งเป็นการประยุกต์ใช้งานจริงในโรงงานอุตสาหกรรมผลิตรถยนต์ (ใช้งานจริงหลายแห่งในสหรัฐอเมริกา) ในเรื่องของการตรวจสอบคุณภาพของสีที่ทาแล้วบนตัวถังรถยนต์ โดยการฉายแสงเลเซอร์ไปยังตัวถังรถยนต์แล้วรับแสงที่สะท้อนมาของสีไปยังจอรับแสง (screen) คุณภาพของแสงที่สะท้อนจะแตกต่างกันไปสำหรับตัวถังที่ทาสีได้เรียบร้อยกับสีที่ทาแล้วคุณภาพไม่ดี ดังแสดงในรูปที่ 5

ระบบดังกล่าวได้ประยุกต์ใช้นิวรอลเน็ตเวอร์กแบบแบคพรอพาเกชัน ดังแสดงโครงสร้างไดอะแกรมของระบบดังรูปที่ 6 ก่อนขั้นตอนของการใช้งานในโรงงานอุตสาหกรรมนี้ นิวรอลเน็ตเวอร์กระบบนี้ได้รับการฝึกฝนโดยได้รับตัวอย่างของภาพอินพุต (ดังแสดงตัวอย่างในรูปที่ 5) จำนวนมาก จนสามารถแยกแยะได้ว่าสัญญาณภาพแบบไหนที่แสดงคุณภาพของสีดีหรือไม่ดี ช่วงขั้นตอนการฝึกฝนนี้ใช้เวลาคำนวณต่อเนื่องโดยไม่หยุดเลยถึง 2 สัปดาห์ (นานมาก)

อย่างไรก็ตาม ในขั้นตอนการใช้งานในโรงงาน ผลลัพธ์ของการฝึกฝน (training หรือ learning) ก็คือ ค่าของน้ำหนักต่าง ๆ ในเน็ตเวอร์ก ผู้สร้างระบบใช้ข้อมูลเหล่านั้นเพื่อเขียนโปรแกรมจำลองแบบ (Simulation) เพื่อให้ระบบทำงานได้บนเครื่องไมโครคอมพิวเตอร์ส่วนบุคคล (ใช้ 80386-PC) ความเร็วในการทำงานในขั้นนี้ไม่ได้เป็นอุปสรรคในการใช้งานจริงเลย



รูปที่ 5 แสดงการสะท้อนของแสงเลเซอร์บนตัวถังที่ทาสีแล้ว
(ก) สีที่ทาแล้วมีคุณภาพไม่ดี
(ข) สีที่ตัวถังรถยนต์มีคุณภาพดี



รูปที่ 6 การใช้งานนิวรอลเน็ตเวอร์คในระบบที่ช่วยตรวจสอบคุณภาพของสีที่ตัวถังรถยนต์ ในอุตสาหกรรมการผลิตรถยนต์

สรุป

ข้อเสียประการสำคัญของนิวรอลเน็ตเวอร์กแบบแบคพรอพาเกชัน (Back-propagation Network) ก็คือความเร็วในขั้นตอนของการเรียนรู้ (Learning Mode) ค่อนข้างช้ามาก โดยเฉพาะอย่างยิ่งเมื่อมีจำนวนของอินพุตและเอาต์พุตจำนวนมาก แต่ถ้าเรียนรู้หรือได้น้ำหนักของจุดเชื่อมโยงเรียบร้อยแล้ว ในโหมดของการใช้งานมักจะไม่ค่อยมีปัญหาเรื่องความเร็ว จึงสามารถนำไปใช้งานในสภาพแวดล้อมจริงได้ดี

นิวรอลเน็ตเวอร์กที่กล่าวถึงในบทความนี้เป็นชนิดที่เรียนรู้จากคู่ตัวอย่างอินพุต เอาต์พุต (Supervised Learning) และเป็นเน็ตเวอร์กแบบหลายชั้นเคลื่อนหน้า (Multilayer Feedforward Network) ยังมีนิวรอลเน็ตเวอร์กในลักษณะแบบอื่น ๆ อีก เช่น นิวรอลเน็ตเวอร์กฮอพฟิลด์ (Hopfield Network) และเครื่องจักรโบลต์ซแมน (Boltzmann Machines) เป็นต้น เน็ตเวอร์กต่าง ๆ เหล่านี้จะแตกต่างกันไปในลักษณะของขั้นตอนการเรียนรู้ วิธีการประมวลผล และประโยชน์ของการใช้งาน

เทคโนโลยีของนิวรอลเน็ตเวอร์กเริ่มแทรกซึมเข้ามามีบทบาทในชีวิตประจำวันของเรามากขึ้นทุกวัน พัดลมหรือตู้เย็นที่บ้านของผู้อ่านบางท่านอาจจะมีลักษณะการทำงานที่ปรับตัวเอง (เรียนรู้) ให้เข้ากับนิสัยของผู้ใช้ได้ โดยการใช้ความสามารถด้านการเรียนรู้ของนิวรอลเน็ตเวอร์กและระบบการควบคุมแบบฟัซซี (Fuzzy Controller) ในตอนต่อไป เราจะมาดูว่าถ้าระบบนิวรอลเน็ตเวอร์กและระบบ Fuzzy Systems มาเจอกัน อะไรจะเกิดขึ้น แล้วเจ้าตัวฟัซซีลอจิก (Fuzzy Logic) มันคืออะไร